An Autonomous Vision-Based Mobile Robot

Eric T. Baumgartner and Steven B. Skaar

Abstract— **This paper describes the theoretical development and experimental implementation of a complete navigation procedure for use in an autonomous mobile robot for structured environments. Estimates of the vehicle's position and orientation are based on the rapid observation of visual cues located at discrete positions within the environment. The extended Kalman filter is used to combine these visual observations with sensed wheel rotations to produce optimal estimates continuously. The complete estimation procedure, as well as the control algorithm, has been developed to be time independent. Rather than time, a naturally suitable quantity involving wheel rotations is used as the independent variable. One consequence of this choice is that the vehicle speed can be specified independently of the estimation and control algorithms.**

Reference paths are "taught" by manually leading the vehicle through the desired path in a manner similar to the teaching of industrial holonomic robots. Estimates produced by the extended Kalman filter during this teaching session are then used to represent the geometry of the path. The tracking of taught reference paths is accomplished by controlling the position and orientation of the vehicle relative to the reference path. Position estimates have been determined to be accurate to within one inch. Also, the reference paths are tracked precisely such that the position error typically does not exceed one inch. Time-independence path tracking has necessitated the development of a novel, geometry-based means for advancing along the reference path. Also novel is a means to accommodate the finite time interval which separates the instant at which images are acquired from the instant as which the processed image information becomes available to update the estimates.

I. Introduction

MANY autonomous mobile robots developed for structured environments rely on guide paths either embedded in or painted on the floor to navigate the robot around the desired workspace [1], [2]. Such mobile robots are adequate for point-to-point tasks where the guide paths do not change over time. Many researchers have realized that mobile robots may need to navigate around their environment without the use of guide paths. Therefore, the accurate estimation of a vehicle's position and orientation relative to its environment becomes critical for precise navigation. Some mobile robots navigate based on position and orientation estimates which are produced by using wheel rotation information alone [3], [4]. This type of estimation has been referred to as dead reckoning. However, dead-reckoned estimates of the position and orientation of the vehicle will be inaccurate over long distances traveled due to imprecisely known initial conditions, errors in the kinematic model of the vehicle, or disturbances during the motion of the vehicle, such as wheel slippage. Therefore, external observations of the surrounding environment must be made during the motion of the vehicle to correct these dead-reckoning errors.

Robotic vehicles have been developed which rely on a visual sensor [5], [6], an optical rangefinder [7], and a sonar or ultrasonic sensor [8]–[10] to observe the surrounding environment. On these vehicles, however, the sensors are used only to correct the dead-reckoned estimates based on the most current measurement set. Each measurement is assumed not to be corrupted with any noise, and, therefore, the accuracy of the estimates is limited by the accuracy of the sensor-based measurements. These methods also assume that the measurement or measurements can fully observe the state of the vehicle, i.e., the position and orientation of the vehicle can be solved for analytically based on the measurement information acquired at a single junction during the motion of the mobile robot.

Recently, a few researchers have realized that new measurements should be weighted according to some scheme which takes into account both the confidence or certainty in the current estimates of the vehicle's position and orientation and the accuracy of the current measurements. Therefore, no one piece of information is limiting, and a combination of past and current measurements is used to estimate optimally the position and orientation of the vehicle. Cox [11] has investigated the use of a laser rangefinder in a structured environment to produce position and orientation estimates via a linear least-squares regression algorithm. Leonard and Durrant-Whyte [12] have also investigated a method to weight incoming measurement information using, instead, a ring of sonar sensors positioned around a mobile robot. Both of these systems assume that an accurate map of the environment is available or that one can be generated [13]. These systems also require that range measurements from the sensors match with the predicted locations of objects within the environment based on the current estimates of the vehicle's position and orientation. When using the above methods in cluttered settings, it is difficult to determine which object within the environment the sensor is actually detecting. Also, proximity-type sensors only provide information about the distance from the sensor to objects in the environment. In a large corridor, for example, mobile robots which rely on proximity sensors will only be provided with information about the vehicle's distance from the corridor walls and the orientation of the vehicle relative to the corridor walls. Both methods described above will not update, and will lose confidence in, the estimates along the corridor unless the end of the corridor can be observed by the proximity sensor.

Manuscript received February 7, 1992; revised April 6, 1993. Recommended by Past Associate Editor R. H. Middleton. This work was supported by the U.S. Office of Naval Research under Contract no. N00014-01-J-1054.

E. T. Baumgartner is with Mechanical Engineering–Engineering Mechanics, Michigan Technological Univerisity, Houghton, MI 49931.

B. Skaar is with the Department of Aerospace and Mechanical Engineering, University of Notre Dame, Indiana 46556.

IEEE Log Number 9213720.

In contrast, the mobile robot presented in this paper depends on a visual sensor to update and correct optimally those position and orientation estimates which are based on wheel-rotation information alone. The extended Kalman filter [14], [15] is applied in this paper to an autonomous mobile robot to weight incoming visual observations of the surrounding environment to produce minimum-variance position and orientation estimates of the vehicle. Observations of the surrounding environment are made via a single camera mounted on the vehicle which detects small, ring-shaped cues which are placed at discrete locations within the environment. These visual cues can be detected rapidly by an image-analysis algorithm every 0.1 seconds. As opposed to the laser rangefinder and the sonar sensor, the visual observation of the wall-mounted cues provides complete information about the position and orientation of the vehicle. Therefore, even in long corridors, both components of translational position as well as orientation will be estimated accurately. The visual cues also provide a distinct signal which is unaffected by other objects within the environment. The reliable and accurate detection of the visual cues produces robust position and orientation estimates even in complex and cluttered settings.

A controller for the autonomous mobile robot has been developed for tracking complex reference paths. Desired reference paths are taught by manually leading the vehicle through the desired path. In an analogous way, fixed-base holonomic manipulators are routinely taught certain paths to follow by manually leading the manipulator through the path or by using a teach pendant to specify points along the path to follows. This teaching procedure is a critical step for performing such tasks as spot welding, assembly, or spray painting in factory environments. Unlike these holonomic robots, where joint rotation is algebraically related to the position and orientation of the end effector, a nonholonomic device such as the robotic vehicle allows at most a differential description of the constraining relationship between internal wheel rotations and the vehicle's position and orientation, and simple repetition of the wheel rotations will not generally result in complex path tracking due to imprecise knowledge of initial conditions or wheel slippage. Therefore, during the teaching of desired reference paths, the extended Kalman filter algorithm is used to produce accurate estimates of the position and orientation of the vehicle. A representation of the taught reference path is then used by the controller for tracking the desired reference path.

The controller developed in this paper also relies on the extended Kalman filter to produce accurate estimates of position and orientation to determine the vehicle's location with respect to the reference path. The normal and angular errors of the vehicle relative to the reference path are then passed through a PID controller to determine the desired velocities to be commanded to the two drive wheels. A novel aspect of the controller is that the speed the vehicle travels along the reference path is specified independent from the path-tracking algorithm. Because of this, the speed of the vehicle can be adjusted in response to any number of possible contingencies. The controller also has the ability to return to the desired reference path following an unexpected departure. Such a departure might be due, for example, to the avoidance of an unanticipated obstacle.

Both the estimation and control algorithms can be applied to either forward or backward maneuvers. Both algorithms, however, require some modifications for backing-up maneuvers. Because the estimation and control algorithms are carried out serially on a single processor, real-time operation is accomplished by interspersing wheel-rotation sensing throughout the image processing algorithm, since this algorithm requires the greatest amount of computational time. Since the extended Kalman filter assumes that measurements are available at the exact instant at which the measurements are sampled, a novel method has been developed for accounting for the finite time between the acquisition of a video image and the identification of cues in that image. This method is necessary because, for the real-time application of the extended Kalman filter presented in this paper, the vehicle generally advances while the image processing takes place.

A novel description of the time-independent differential kinematics of the vehicle is presented in Section II, while Section III describes the vision-based estimation algorithm used to produce optimal estimates of the vehicle's position and orientation. Section III also presents the method developed for accounting for the interval between the acquisition of an image and the detection of a visual cue in the acquired image. The method used for teaching desired reference paths is described in Section IV while the controller developed for tracking the desired reference paths is presented in Section V. Experimental results concerning the accuracy of the vehicle's position and orientation estimates produced by the extended Kalman filter and the tracking accuracy of the controller are presented in Section VI.

II. Vehicle Kinematics

Many kinematic models describing the motion of various vehicle configurations have been developed [3], [7], [11]. These descriptions usually involve expressing the time derivatives of the vehicle's position and orientation in terms of time-based quantities such as the rotational speed of the vehicle's wheels. Recently, vehicle kinematics which are independent of time have been derived for a vehicle with a tricycle configuration [16], [17]. A similar kinematic description is presented in this paper for the drive configuration pictured in Fig. 1, where two front wheels are driven independently, producing both translational and rotational control of the vehicle. The back wheels consist of two free-to-rotate castor wheels. For the vehicle configuration presented in this paper, as shown schematically in Fig. 2, an independent variable, α, can be defined as the average forward rotation of the two front drive wheels, while a control variable, u, can be defined as the difference between the differential rotations of the front two drive wheels normalized by the sum of the differential rotations of the front two drive wheels, i.e.,

$$\alpha = \frac{\theta_r + \theta_l}{2} \tag{2.1}$$

$$u = \frac{d\theta_r - d\theta_l}{d\theta_r + d\theta_l} \tag{2.2}$$

Fig. 1. Autonomous vision-based mobile robot.

where θ_r and θ_l represent the forward rotations of the right and left drive wheels, respectively, and $d\theta_r$ and $d\theta_l$ represent the differential forward rotations of the right and left drive wheels. For this choice of the independent variable and the control variable, kinematic state equations of the vehicle can be written in the following state space form

$$\underline{x}'(\alpha) = \frac{d\underline{x}(\alpha)}{d\alpha} = \begin{bmatrix} dX(\alpha)/d\alpha \\ dY(\alpha)/d\alpha \\ d\phi(\alpha)/d\alpha \end{bmatrix} = \underline{f}(\underline{x}(\alpha), u(\alpha)) \quad (2.3)$$

where $\underline{x}$ represents the three-element state vector which for this problem is defined by the two translational coordinates of a selected point on the vehicle (X, Y) and the in-plane orientation of the vehicle (ϕ) as shown in Fig. 2, and where $\underline{f}$ represents a set of three nonlinear functions of the state. For example, if the selected point on the vehicle is chosen as the midpoint between the two front drive wheels (i.e., the center of rotation of the vehicle), then the kinematic state equations would be

$$\underline{x}'(\alpha) = \begin{bmatrix} dX(\alpha)/d\alpha \\ dY(\alpha)/d\alpha \\ d\phi(\alpha)/d\alpha \end{bmatrix} = \begin{bmatrix} R\cos\phi(\alpha) \\ R\sin\phi(\alpha) \\ \frac{R}{b}u(\alpha) \end{bmatrix} \quad (2.4)$$

where R represents the radius of the drive wheels and b represents half the distance between the drive wheels. This kinematic description is independent of time.

If precise initial conditions of the position and orientation of the vehicle are known, then the above kinematic state equations (2.3) can be integrated numerically or "dead-reckoned" by determining the independent variable, α, and the control variable, u, using information from two optical shaft encoders, one for each wheel. Truncation errors due to the finite precision of any numerical integration scheme, disturbances during the motion of the vehicle (e.g., wheel slippage), or inaccuracies in the kinematic model of the vehicle will cause errors in the position and orientation estimates of the vehicle to grow over long distances. The nature of the estimation method

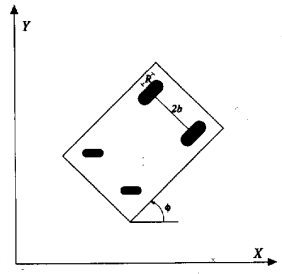


Fig. 2. Vehicle schematic.

presented in the following section is to correct any errors in the integrated or dead-reckoned estimates of the vehicle's position and orientation via visual observations of the surrounding environment as they become available.

III. Vision-Based Estimation

The autonomous vehicle uses a visual sensor to view the surrounding environment and to update and correct the dead-reckoned estimates of the vehicle's position and orientation. If detectable cues are placed or located at known positions within the operating workspace of the vehicle, then the location of detected cues in the image plane of the video camera mounted on the vehicle will provide some information which is related to the position and orientation of the vehicle within the workspace. Cues which may be recognizable from discrete video data include simple ring shapes (which are used in the experiments reported herein and have been used extensively in the positioning of holonomic and nonholonomic robots [16], [18]), electric wall outlets, corners of rooms and doorways. If the computational time necessary for detecting these cues remains manageable, then vision can be used to update the position and orientation estimates of the vehicle in real-time.

An observation equation, which relates a measurable quantity to the state of the vehicle, can be determined for the single visual sensor placed on the vehicle. Using a pin-hole model for the video camera [19], the horizontal location of a cue in the image plane of the camera, x_c, (measured in pixels) is related to the three-element state vector via the following nonlinear algebraic relationship

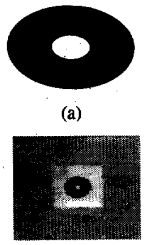
$$x_c = C_1 \frac{X\cos(\phi + C_4) + Y\sin(\phi + C_4) - C_2}{X\sin(\phi + C_4) - Y\cos(\phi + C_4) - C_3} \quad (3.1)$$

where C_1, C_2, C_3, and C_4 are calibration parameters which depend in part on the focal length of the camera and the position and orientation of the camera on the vehicle. This relationship can be expressed in the general form

$$z(\alpha_a) = h(\underline{x}(\alpha_a)) \quad (3.2)$$

where z represents the measurement, h represents a nonlinear observation function, and α_a is the value of the independent variable at the instant at which an image is acquired.

An algorithm for rapidly detecting the centroid of ring shapes [18], which are used in this problem as the visual cues, is used both to calibrate the video camera and to update estimates during real-time operation. A typical ring-shaped



(a)

(b)

Fig. 3. Visual cues.

cue, and the view of this cue in the image plane of a video camera mounted on the vehicle, is shown in Fig. 3. The cues may be either white centered or black centered. All cues, however, are constructed such that the diameter of the central ellipse is equal to the width of the surrounding ring. Detection of such a cue is based on the simple geometric fact that a circular or elliptical cue, viewed from any perspective, is nearly elliptical in the image plane of the camera. Thus, any line through the center of the cue will cut the cue in symmetric lengths of alternating black and white. The cue detection algorithm scans the image plane for patterns which result in symmetric lengths of black and white. When this pattern is identified, the algorithm searches in three more directions (one vertical, two diagonal) to confirm that the pattern is indeed a ring-shaped cue. Rapid and reliable detection of the cues via this algorithm has been achieved as one cue can be detected on the order of once every 0.1 seconds using a 80386-based processor.

The calibration parameters in (3.1) are determined in a least-squared-error sense using Marquardt's method [20] by detecting the horizontal location of the cue in the image plane of the camera at known positions and orientations of the vehicle. The resultant calibration parameters for the current configuration of the camera on the vehicle are given in Table I. Fig. 4 shows the measured and best-fit horizontal locations of the cues in the image plane of the video camera for each of the 96 cue observations used in the calibration procedure. The best-fit image-plane horizontal locations of the cues are determined by substituting the measured positions and orientations of the vehicle into (3.1) using the optimal calibration parameters given in Table I. The mean of the absolute value of the error between the measured and best fit cue locations for this calibration set is 2.601 pixels with a standard deviation of 1.972 pixels. The mean of the error for this calibration set represents only 0.51% of the total image plane resolution of 512 pixels.

The kinematic state equations (2.3) and the observation equation (3.2) have the required form for the application of a standard state estimation algorithm to determine the position and orientation estimates of the vehicle. The extended Kalman filter for the continuous-discrete problem [14], [15] is applied to this system. This algorithm produces minimum mean-square-error estimates when the noise associated with the state equations, known as process noise, $\underline{w}(\alpha)$, is assumed to be a zero-mean, Gaussian distributed, white noise process with a known covariance matrix, $Q(\alpha)$. The noise associated with the

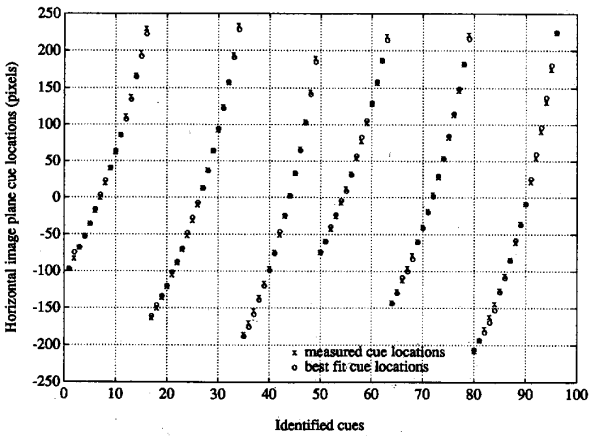


Fig. 4. Camera calibration results.

TABLE I
CAMERA CALIBRATION PARAMETERS

Parameter	Calibration Value
C_1	767.67 pixels
C_2	4.07 inches
C_3	8.94 inches
C_4	2.29 radians

observation equations, known as measurement noise, $\underline{v}(\alpha_a)$, must be assumed to be a zero-mean, Gaussian, white process with a known covariance matrix, $R(\alpha_a)$. The process noise and the measurement noise are assumed to be uncorrelated. Thus, consider a system described by state equations and observation equations of the form

$$\underline{x}'(\alpha) = \underline{f}(\underline{x}(\alpha), u(\alpha)) + \underline{w}(\alpha) \tag{3.3}$$

$$\underline{z}(\alpha_a) = \underline{h}(\underline{x}(\alpha_a)) + \underline{v}(\alpha_a). \tag{3.4}$$

Clearly for the system presented in this paper, the process noise and the measurement noise will not be a Gaussian, white noise processes. Nonetheless, Section VI will show that estimates produced by the extended Kalman filter are physically accurate.

When observations are not available, the nonlinear kinematic state equations (2.3) are used to propagate the estimates of the state, denoted by $\underline{\hat{x}}(\alpha)$, by integrating numerically the state equations (i.e., dead reckoning) via a fourth order Runge–Kutta integration scheme [20]. A linearization of the state equations about the current estimate of the state is used to propagate the estimation error covariance matrix of the state, denoted by $P(\alpha)$. When observations become available, the extended Kalman filter algorithm uses a linearlization of the observation equations (3.1) about the current estimate of the state to update both the state and its estimation error covariance matrix. The extended Kalman filter algorithm proceeds according to the following three steps.

1) The state estimates, $\underline{\hat{x}}(\alpha)$, and the estimation error covariance matrix, $P(\alpha)$, are initialized at $\alpha = 0$.

2) The state estimates and the estimation error covariance matrix are propagated when observations are not available by integrating numerically the following differential equations:

$$\underline{\hat{x}}'(\alpha) = \underline{F}(\underline{\hat{x}}(\alpha), u(\alpha)) \tag{3.5}$$

$$P'(\alpha) = f(\underline{\hat{x}}(\alpha), u(\alpha))P(\alpha) + P(\alpha)F^T(\underline{\hat{x}}(\alpha), u(\alpha)) + Q(\alpha) \tag{3.6}$$

where

$$F(\underline{\hat{x}}(\alpha), u(\alpha)) = \left.\frac{\partial \underline{f}(\underline{x}, u)}{\partial \underline{x}}\right|_{\substack{\underline{x}=\underline{\hat{x}}(\alpha) \\ u=u(\alpha)}}. \tag{3.7}$$

3) The state estimates and the estimation error covariance matrix are then updated when observations are acquired at α_a:

$$\underline{\hat{x}}(\alpha_a|\alpha_a) = \underline{\hat{x}}(\alpha_a) + K(\alpha_a)[\underline{z}(\alpha_a) - \underline{h}(\underline{\hat{x}}(\alpha_a))] \tag{3.8}$$

$$P(\alpha_a|\alpha_a) = [I - K(\alpha_a)H(\underline{\hat{x}}(\alpha_a))]P(\alpha_a) \tag{3.9}$$

$$K(\alpha_a) = P(\alpha_a)H^T(\underline{\hat{x}}(\alpha_a)) \cdot [H(\underline{\hat{x}}(\alpha_a))P(\alpha_a)H^T(\underline{\hat{x}}(\alpha_a)) + R(\alpha_a)]^{-1} \tag{3.10}$$

where

$$H(\underline{\hat{x}}(\alpha_a)) = \left.\frac{\partial \underline{h}(\underline{x})}{\partial \underline{x}}\right|_{\underline{x}=\underline{\hat{x}}(\alpha_a)} \tag{3.11}$$

and $\underline{\hat{x}}(\alpha_a)$ and $P(\alpha_a)$ are the propagated state estimates and the propagated estimation error covariance matrix, respectively, which are determined by Step 2 just prior to the acquisition of an observation at α_a.

Note that the independent variable, α, for this filter can run either forward or backward depending on whether the vehicle is proceeding forward or backing up. Since the extended Kalman filter normally is applied to systems where time is the independent variable, the case where time runs backwards is not normally considered. For the application presented in this paper, however, this case must be considered. The propagation of the state estimates (3.3) is valid for forward or backward maneuvers. To make the filter for backward maneuvers equivalent to the forward filter, however, the propagation of the estimation error covariance via (3.4) must be modified. For backing-up maneuvers, (3.4) must be modified as follows:

$$P'(\alpha) = F(\underline{\hat{x}}(\alpha), u(\alpha))P(\alpha) + P(\alpha)F^T(\underline{\hat{x}}(\alpha), u(\alpha)) - Q(\alpha) \tag{3.12}$$

where this relationship is derived by setting $\alpha = -\alpha$ in (3.4) and evaluating the result. All other aspects of the filter remain the same.

In conventional Kalman filter applications, the measurements or observation are assumed to be available at the exact instant at which they are acquired. For the real-time vision-based application presented in this paper, this assumption is not valid due to the finite computational time needed to detect a visual cue in an acquired video image. In general the independent variable, α, advances (i.e., the vehicle moves) while the image processing is taking place, and the detection of a visual cue does not occur until some finite value of α after the image has been acquired. Therefore, a novel method is developed below to transition the relevant information from the visual measurements from the value of α at which the video image is acquired to the value of α at which the cues in the video image are actually identified.

If α_a represents, as before, the value of α at which the video image is acquired and α_p represents the value of α at which the image processing is complete and cues in the image are identified, then the estimates of the state at α_p can be updated with the cue information acquired by the video image taken at α_a by the following linear relationship

$$\underline{\hat{x}}(\alpha_p|\alpha_a) = \underline{\hat{x}}(\alpha_p) + \Delta\underline{\hat{x}}(\alpha_p) \tag{3.13}$$

where $\underline{\hat{x}}(\alpha_p|\alpha_a)$ denotes the state estimate at α_p given the observations up to α_a and where $\underline{\hat{x}}(\alpha_p)$ is propagated up to α_p via the kinematic state equations (3.5). Taking derivatives of (3.13) with respect to α_p yields

$$\underline{\hat{x}}'(\alpha_p|\alpha_a) = \underline{\hat{x}}'(\alpha_p) + \Delta\underline{\hat{x}}'(\alpha_p) \tag{3.14}$$

however,

$$\begin{aligned}\underline{\hat{x}}'(\alpha_p|\alpha_a) &= \underline{f}(\underline{\hat{x}}(\alpha_p|\alpha_a), u(\alpha_p)) \\ &= \underline{f}(\underline{\hat{x}}(\alpha_p) + \Delta\underline{\hat{x}}(\alpha_p), u(\alpha_p)) \\ &= \underline{f}(\underline{\hat{x}}(\alpha_p), u(\alpha_p)) + \left.\frac{\partial \underline{f}(\underline{x}, u)}{\partial \underline{x}}\right|_{\substack{\underline{x}=\underline{\hat{x}}(\alpha_p) \\ u=u(\alpha_p)}} \\ &\quad \cdot \Delta\underline{\hat{x}}(\alpha_p) + \cdots\end{aligned} \tag{3.15}$$

and

$$\underline{\hat{x}}'(\alpha_p) = \underline{f}(\underline{\hat{x}}(\alpha_p), u(\alpha_p)). \tag{3.16}$$

Substituting (3.15) and (3.16) into (3.14) and keeping only first-order terms yields

$$\Delta\underline{\hat{x}}'(\alpha_p) = F(\underline{\hat{x}}(\alpha_p), u(\alpha_p))\,\Delta\underline{\hat{x}}(\alpha_p) \tag{3.17}$$

where

$$F(\underline{\hat{x}}(\alpha_p), u(\alpha_p)) = \left.\frac{\partial \underline{f}(\underline{x}, u)}{\partial \underline{x}}\right|_{\substack{\underline{x}=\underline{\hat{x}}(\alpha_p) \\ u=u(\alpha_p)}}. \tag{3.18}$$

The solution of (3.17) in general is given by

$$\Delta\underline{\hat{x}}(\alpha) = \Phi(\alpha, \alpha_0)\,\Delta\underline{\hat{x}}(\alpha_0) \tag{3.19}$$

where the state transition matrix, $\Phi(\alpha, \alpha_0)$, is found by solving

$$\Phi'(\alpha, \alpha_0) = F(\underline{\hat{x}}(\alpha), u(\alpha))\Phi(\alpha, \alpha_0) \tag{3.20}$$

with the initial condition $\Phi(\alpha_0, \alpha_0) = I$ where I is the $n \times n$ identity matrix (for this problem, $n = 3$). Therefore, from (3.13), the state estimate at α_p given the observations up to α_a, $\underline{\hat{x}}(\alpha_p|\alpha_a)$, is given by

$$\begin{aligned}\underline{\hat{x}}(\alpha_p|\alpha_a) &= \underline{\hat{x}}(\alpha_p) + \Delta\underline{\hat{x}}(\alpha_p) \\ &= \underline{\hat{x}}(\alpha_p) + \Phi(\alpha_p, \alpha_a)\,\Delta\underline{\hat{x}}(\alpha_a) \\ &= \underline{\hat{x}}(\alpha_p) + \Phi(\alpha_p, \alpha_a)[\underline{\hat{x}}(\alpha_a|\alpha_a) - \underline{\hat{x}}(\alpha_a)] \\ &= \underline{\hat{x}}(\alpha_p) + \Phi(\alpha_p, \alpha_a)K(\alpha_a) \\ &\quad \cdot [\underline{z}(\alpha_a) - \underline{h}(\underline{\hat{x}}(\alpha_a))]\end{aligned} \tag{3.21}$$

where the state transition matrix in (3.21) is propagated in real time by integrating numerically (3.20) from α_a to α_p. Note the similarity of (3.21) to the update equation (3.8) associated with the conventional extended Kalman filter. In (3.21), the change in the estimate due to new observations at α_a is transitioned via the state transition matrix to α_p. A similar method is used to update the estimation error covariance matrix, P, at α_p using the observations at α_a.

IV. Reference Path Teaching

Based on the position and orientation estimates produced by the extended Kalman filter algorithm presented in Section III, desired reference paths to be followed are 'taught' to the vehicle. A 'teach–repeat' mode is typically used in the control of holonomic or fixed-based robotic systems which are used extensively in industrial applications. A desired path is taught to the holonomic robot by using a teach pendant to record the joint-rotation sequences of settings which result in the completion of the desired task. The task is then repeated by playing back the taught joint rotations. For holonomic systems, the joint rotations are algebraically related to the position and orientation of any point on the robot. Therefore, a sequence of joint rotation settings will result in the same position and orientation of the point on the robot independent from the order in which the sequence of joint rotations is repeated and independent of the way in which the joints transition from setting to setting. For a nonholonomic robot, such as the mobile robot, however, the wheel (joint) rotations are at most differentially related to the position and orientation of the robot as shown in Section II, and the position and orientation of the mobile robot is dependent on the way in which the sequence of wheel rotations is repeated. Likewise, a different terminal position and orientation of the vehicle will be achieved for the same joint rotation sequence if, for example, the initial conditions for the vehicle are different for successive runs of the vehicle.

To achieve a teach–repeat mode similar to the holonomic robot, the nonholonomic system described in this paper is taught by guiding the vehicle through the desired path. During the teaching procedure, estimates of the vehicle's position and orientation are generated. The taught path is then saved in a manner which is compatible with a tracking procedure which, in turn, is used to repeat the taught path. This teaching procedure holds many practical advantages over other alternative methods where paths are planned based on the complete floor plan of the environment. In practice, knowledge of this type is not enough to ensure precise navigation of the vehicle especially when approaching an area which requires tight-tolerance maneuvers (e.g., passing through a door). Also, motion which is required to bring the vehicle to a desired position and orientation may involve a complex series of maneuvers (i.e., backing up and going forward again). Such maneuvers may be difficult to achieve by means of an artificial path planning algorithm. The use of a teacher allows the judgment of the teacher to be invoked. Humans are very adept at controlling nonholonomic systems, and the use of a human to teach paths provides a high level of path planning capability which is otherwise difficult to achieve.

As shown in the next section, the control of the vehicle when tracking the desired reference paths is independent of time, and, thus, the rate at which the vehicle travels along the path is specified independent from the path tracking algorithm. Therefore, let the desired reference path be expressed as functions of the arc length of the path, i.e., let $X_{\text{ref}} = X_{\text{ref}}(s)$, $Y_{\text{ref}} = Y_{\text{ref}}(s)$, and $\phi_{\text{ref}} = \phi_{\text{ref}}(s)$ where s represents the arc length of the path. For example, the entire reference path or a segment of the reference path can be specified by the following polynomial functions:

$$X_{\text{ref}}(s) = \sum_{l=1}^{L} a_l s^l \tag{4.1}$$

$$Y_{\text{ref}}(s) = \sum_{l=0}^{L} b_l s^l \tag{4.2}$$

$$\phi_{\text{ref}}(s) = \sum_{l=0}^{L} c_l s^l \tag{4.3}$$

where a_l, b_l, and c_l are polynomial coefficients and L is the desired order of the polynomial functions. The estimation sequence generated by the extended Kalman filter during the teaching of the reference path is then used to fit the above polynomial functions to the entire path or to segments of the path. Typically, linear functions of the arc length (i.e., $L = 1$) are used to represent segments of the desired reference path. The reference path parameters a_0, a_1, b_0, b_1, c_0, and c_1 are then determined by fitting a straight line over a group of position and orientation estimates produced by the extended Kalman filter.

The above procedure allows the desired reference path to be expressed in terms of the specific workspace. In other words, small absolute errors which may occur in the placement or the location of the cues in the workspace will be included in a certain sense within the taught reference path. If the workspace is large, the positioning or locating of visual cues will become less accurate in absolute terms as the distance of the cues from the origin of the coordinate system increases. Therefore, position and orientation estimates of the vehicle based on the extended Kalman filter are generated in relation to that subset of cues which appear in the image plane of the vehicle's video sensor in any particular region of operation. Even though the estimates may not be physically accurate in a global sense, the estimates will be locally accurate in terms of their relationship to one another. Thus, desired reference paths which are taught based on current estimates of the position and orientation of the vehicle will be locally accurate. Therefore, the tracking of these reference paths will also be locally accurate, which is the desired performance characteristic.

V. Path Tracking and Control

Many conventional control systems proposed for autonomous robotic vehicles command a series of reference positions and orientations at a fixed time interval. These autonomous vehicles must control both the translational and

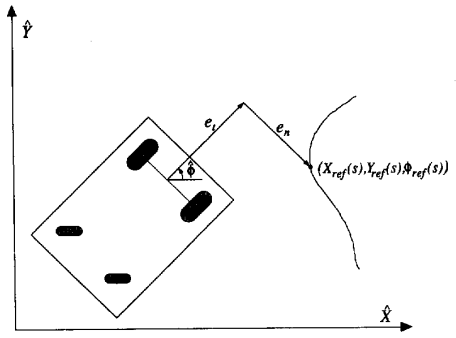


Fig. 5. Vehicle navigation and control.

rotational velocities of the vehicle to follow the desired reference path [3], [11]. If the autonomous vehicle falls behind the reference sequence, the time-based controller will command the vehicle to miss certain sections of the reference path to 'catch up' with the commanded reference sequence. For example, large errors created by an obstacle-avoidance maneuver would cause the vehicle to fall behind the time-based reference sequence. The autonomous robotic vehicle presented in this paper is required to track precisely the desired reference path rather than simply arrive at a prescribed location. Thus, a time-based controller would be incompatible with this objective.

As shown in Fig. 5, navigation of the vehicle can be based upon tangential, normal, and angular errors with respect to the desired reference path which is taught as discussed in the previous section. The errors between a point on the vehicle and a point on the reference path in the tangential, normal, and angular directions are given by

$$e_t = (X_{\text{ref}}(s) - \hat{X}(\alpha))\cos\hat{\phi}(\alpha) + (Y_{\text{ref}}(s) - \hat{Y}(\alpha))\sin\hat{\phi}(\alpha) \quad (5.1)$$

$$e_n = (X_{\text{ref}}(s) - \hat{X}(\alpha))\sin\hat{\phi}(\alpha) - (Y_{\text{ref}}(s) - \hat{Y}(\alpha))\cos\hat{\phi}(\alpha) \quad (5.2)$$

$$e_\phi = \phi_{\text{ref}}(s) - \hat{\phi}(\alpha) \quad (5.3)$$

where $\hat{X}(\alpha)$, $\hat{Y}(\alpha)$, and $\hat{\phi}(\alpha)$ represent estimates of the position and orientation of the vehicle as determined by the extended Kalman filter algorithm described in Section III. For time-based controllers, the error in the tangential direction, e_t, is used to control the speed that the vehicle travels along the reference path [3]. Instead, for the control algorithm presented in this paper, the error in the tangential direction is used to determine the juncture along the reference path which is to be used as the reference point for control. This novel approach allows for the speed of the vehicle as it proceeds along the reference path to be specified independent from the tracking of the reference path.

At any instant, estimates of the position and orientation of the vehicle are known. Therefore the only unknowns in (5.1) are e_t and s. For example, if one chooses $e_t = 0$ in (5.1) and $L = 1$ in (4.1) and (4.2), then (5.1) yields

$$e_t = 0 = (a_0 + a_1 s - \hat{X}(\alpha))\cos\hat{\phi}(\alpha) + (b_0 + b_1 s - \hat{Y}(\alpha))\sin\hat{\phi}(\alpha) \quad (5.4)$$

where a_0, a_1, b_0, and b_1 are known from the reference path teaching procedure described in the previous section. Solving (5.4) for s yields

$$s = \frac{-[(a_0 - \hat{X}(\alpha))\cos\hat{\phi}(\alpha) + (b_0 - \hat{Y}(\alpha))\sin\hat{\phi}(\alpha)]}{[a_1\cos\hat{\phi}(\alpha) + b_1\sin\hat{\phi}(\alpha)]} \quad (5.5)$$

where s is the value of the arc length along the reference path of the point to be used for control purposes. Once the value of s is known, the error in the normal direction, e_n, and the angular error, e_ϕ, are determined via (5.2) and (5.3), respectively. By choosing $e_t = 0$, the value of s given by (5.5) yields the point along the reference path which is directly normal to the vehicle.

The commanded control variable defined in (2.2) is then determined by passing the error in the normal direction and the error in the angular direction through a PID controller

$$u = k_1 e_n + k_2\int^{\alpha} e_n\, d\alpha + k_3\frac{de_n}{d\alpha} + k_4 e_\phi + k_5\int^{\alpha} e_\phi\, d\alpha + k_6\frac{de_\phi}{d\alpha} \quad (5.6)$$

where k_1, k_2, k_3, k_4, k_5, and k_6 are chosen to produce the desired closed-loop response. The rotational velocities commanded to the right and left front driving wheels, $\dot{\theta}_r$ and $\dot{\theta}_l$, respectively, are then found by the following relationships:

$$\dot{\theta}_r = \frac{d\theta_r}{dt} = \frac{d\alpha}{dt}\frac{\theta_r}{d\alpha} = \dot{\alpha}_{\text{ref}}(1 + u) \quad (5.7)$$

$$\dot{\theta}_l = \frac{d\theta_l}{dt} = \frac{d\alpha}{dt}\frac{d\theta_l}{d\alpha} = \dot{\alpha}_{\text{ref}}(1 - u) \quad (5.8)$$

where $\dot{\alpha}_{\text{ref}}$ represents the separately-determined reference translational speed of the vehicle. The above procedure for controlling the motion of the vehicle is based only on the position and orientation of the vehicle with respect to the desired reference path. Therefore, the control of the vehicle is independent of time because the reference point is computed while the vehicle is in motion rather than being an *a priori* function of time.

The reference translational speed of the vehicle, $\dot{\alpha}_{\text{ref}}$, can be specified based on the type of motion desired, such as accelerated, constant speed, or decelerated motion. The control strategy presented in this section allows the speed to be altered for a variety of reasons. For example, if the certainty in the estimates decreases during tracking, the forward speed may be reduced so that more visual cues can be observed to increase estimation accuracy. The vehicle's speed may also affect the accuracy of the path tracking due to inertial effects, and the speed may be reduced during complex portions of the path. In other words, the translational speed of the vehicle could be increased or decreased due to the simplicity or complexity of the desired reference path.

As stated in Section III, backing-up maneuvers require a modification to the estimation algorithm. They also require some modification to the control algorithm presented in this section. The stability of the controller is affected by the point on the vehicle in terms of which the translational position is described [21]. It turns out that any point on the vehicle which is forward of the two drive wheels is a stable point in terms of control if the vehicle is moving forward. For backing-up maneuvers, however, any point behind the two drive wheels results in control stability. Any point along the line joining the two drive wheels is suitable for either forward or backward maneuvers. Also, because of the definition of the control variable, the sign of the control gains differs for forward and backward maneuvers. For forward maneuvers, the control gains operating on the error in the normal direction, k_1, k_2, and k_3, should all be negative while the control gains operating on the error in angular direction, k_4, k_5, and k_6, should all be positive. For backing-up maneuvers, the control gains operating on the error in the normal direction should remain negative; the control gains operating on the error in angular direction, however, should reverse in sign and should be negative.

VI. Experimental Results

The autonomous vehicle pictured in Fig. 1 carries an 80386-based personal computer to perform the image analysis, estimation of the position and orientation of the vehicle, and reference path teaching and tracking, all in real-time. A single standard closed-circuit television video camera is used for observing the wall cues. Within the personal computer, a frame-grabber board is used for image acquisition while a two-axis motor controller board is used both for sampling the wheel rotations and for commanding the drive wheel velocities required for path tracking. Optical shaft encoders are used for sensing the drive wheel rotations. Currently the vehicle is tethered for electric power alone. Estimates of the position and orientation of the vehicle are produced every 40 milliseconds while the estimates are updated via new observations of visual cues every 0.1 seconds.

The experimental workspace for the autonomous vehicle is shown in Fig. 6. Ten cues are spaced around the workspace as shown in Fig. 6 approximately one foot from the floor. To investigate the ability of the state estimation algorithm to produce accurate position and orientation estimates of the vehicle, the vehicle was physically pushed through a series of paths and stopped at various junctures. At these junctures, an image was recorded and the centroids of any cues appearing in the image plane of the video camera were detected. The extended Kalman filter algorithm presented in Section III was used to produce estimates of the position and orientation of the vehicle. The position estimates were then compared with the measured or actual position of the vehicle. The errors between the actual and estimated position estimates in the X and Y direction were then computed. Table II shows the average of the absolute value of the errors between the actual and estimated vehicle positions for various trials. This table shows that estimation errors in the position of the vehicle do not exceed one inch. Although the orientation of the vehicle was not measured, orientation estimates produced by the extended Kalman filter appeared to be accurate as well.

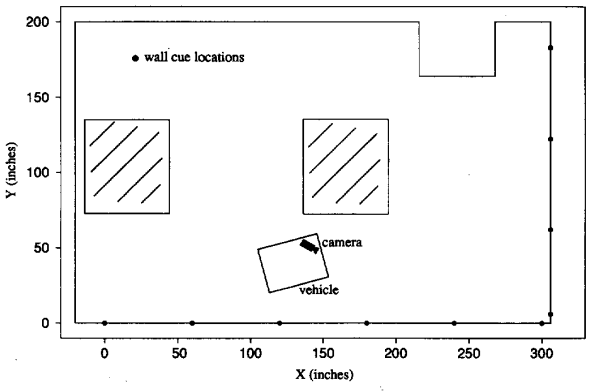


Fig. 6. Experimental workspace.

TABLE II
Average Estimation Error

Trial	Average estimation error in the X direction (inches)	Average estimation error in the Y direction (inches)
1	0.596	0.817
2	0.386	0.193
3	0.240	0.321
4	0.333	0.488

Since the autonomous vehicle has the ability to estimate accurately its position and orientation within a given workspace, complex desired reference paths can be taught as outlined in Section IV. The vehicle was taught a path to follow by manually leading it through a path and by generating the desired reference path in the form of (4.1)–(4.3). The estimation algorithm described in Section III was used to determine estimates of the position and orientation of the vehicle during this teaching mode. Groups of 15 estimates were then used to fit linearly (i.e., $L = 1$ in (4.1)–(4.3)) these estimates to one segment of the reference path. The segments of a desired reference path to be tracked by the vehicle are shown in Fig. 7. The entire reference path shown in Fig. 7 includes 146 segments. This limited amount of data can be stored easily within the onboard computer during the tracking of the desired reference path. Note that the desired reference path requires the vehicle to back up. This maneuver occurs at approximately $Y = 70$ inches and between $X = 75$ and 125 inches. Also note that the desired reference path avoids the large obstacle in the middle of the workspace. Since the desired reference path was taught by a human operator who will undoubtedly avoid anticipated obstacles, the vehicle when tracking the desired path will also avoid anticipated obstacles.

Fig. 8 shows the autonomous vehicle tracking the desired reference path. To begin the tracking procedure, the vehicle was placed approximately in the desired starting location. Once the vehicle starts is motion and the estimation procedure begins, any errors in the initial placement of the vehicle or errors in the assumed initial conditions will be corrected in

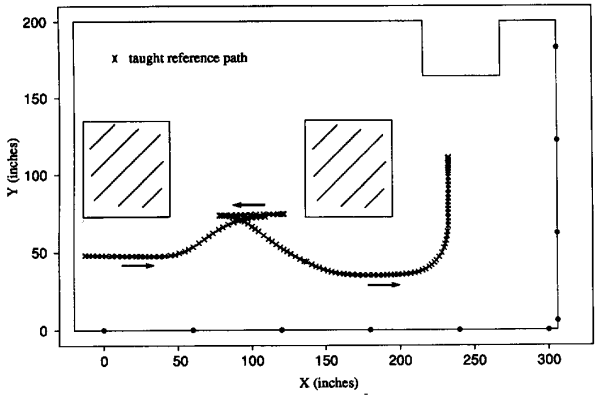


Fig. 7. Taught reference path.

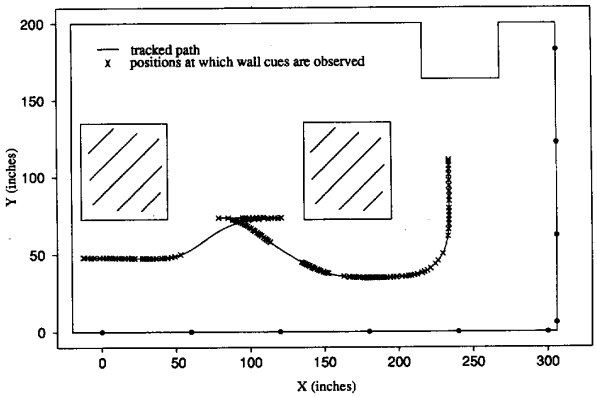


Fig. 9. Tracked path with location of cue observations.

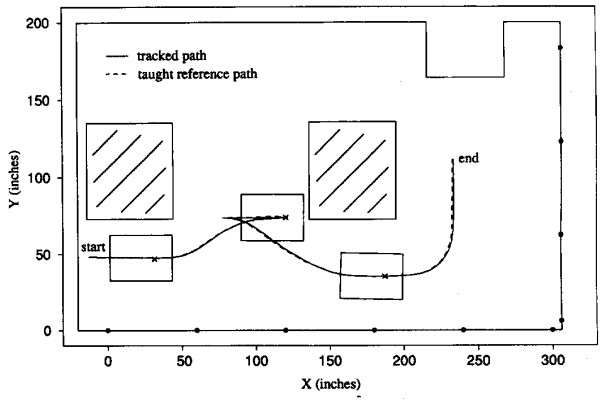


Fig. 8. Tracked path.

the estimation process. Fig. 9 shows the tracked path again, along with those vehicle positions where a cue centroid was detected by the image processing algorithm. Note that in some regions of the tracked path, no cues appear in the image plane of the video camera. In these regions, position and orientation estimates of the vehicle are based on dead reckoning alone. Once cues again appear in the image plane of the camera mounted on the vehicle, they are included in the position and orientation estimates of the vehicle. The vehicle tracked the taught reference path such that the maximum error in the normal direction, e_n, never exceeded one inch.

To complete the backing-up maneuver, the vehicle must decelerate before stopping, then accelerate in the opposite direction. This maneuver can be accomplished easily using the control strategy outlined in Section V. Since the reference translational speed of the vehicle can be specified independent from the controller, changes in the vehicle's speed can be specified *a priori* using, for example, knowledge of the locations along the reference path where direction changes will occur.

The control gains in (5.4), $k1 - k6$, were chosen such that the closed-loop controller of the vehicle tracked the desired reference path in a smooth and stable manner. Both experimental results and a simulation of the closed-loop controller have shown that unstable motion can be obtained for certain choices of the control gains. Along with the choice of control gains, the estimation procedure also affects the stability of closed-loop performance. When large residuals occur between the predicted locations of the visual cues and the actual observed locations of the visual cues, large changes in the estimates of the position and orientation of the vehicle may also occur. These changes in the estimates which are based on visual observations can effect the closed-loop stability of the controller. If, for example, large discontinuities in the position and orientation estimates of the vehicle occur, then the error in the direction normal to the desired reference path, e_n, and the error in orientation from the desired reference path, e_ϕ, will also be large resulting in a large value of the commanded control variable given by (5.4). Such large values of the commanded control have been observed to destabilize the closed-loop controller.

Finally, the choice of the reference translational speed of the vehicle, $\dot{\alpha}_{\text{ref}}$, also affects the stability of the closed-loop controller. Cox [11] has suggested that the control gains should be scaled inversely by the translational speed of the vehicle, thus reducing the commanded control variable when the vehicle is moving fast. A similar method for scaling the control variable according to the translational speed of the vehicle has been implemented on the vehicle presented in this paper. Recently a linear performance and stability analysis of the entire closed loop control system for the nonholonomic vehicle has been completed [21]. This analysis has verified the above experimental findings concerning the choice of the PID control gains and the choice of the reference speed of the vehicle.

VII. Conclusion

A method for combining current and past observations of visual cues with wheel rotation information to produce optimal position and orientation estimates has been presented. The estimation procedure is carried out via the extended Kalman filter for a system which uses the average of the two forward drive wheel rotations as the independent variable. The filter continuously produces accurate position and orientation estimates throughout a structured environment. A novel method

has been developed which accounts for the finite time interval between the acquisition of a visual image and the detection of cues in that image. Desired reference paths are taught based on the sequence of accurate estimates. A closed-loop controller is also presented which precisely tracks complex reference paths.

Future work includes developing image analysis algorithms for detecting naturally occurring features in the environment for use as visual cues. A second camera which will increase the chance of detecting cues not present in the image plane of the other camera will be added to the vehicle to increase the workspace of the vehicle. While no provision has yet been made for an unanticipated obstacle avoidance capability, the tracking algorithm presented in this paper is inherently compatible with such a capability because of its time-independence and its ability to return to the reference path following a large, unexpected departure.

ACKNOWLEDGMENT

The authors would like to thank Dr. Richard K. Miller, distinguished professor of Mathematics at Iowa State University, for his contribution to the development of the image analysis algorithm.

REFERENCES

[1] M. H. E. Larcommbe, "Tracking stability of wire guided vehicles," in *Proc. Internat. Conf. Automatically Guided Vehicle Systems,* Jun. 1981. pp. 137–144.

[2] K. C. Drake, E. S. McVey, and R. M. Inigo, "Sensing error for a mobile robot, using line navigation," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 7, pp. 485–490, Jul. 1985.

[3] Y. Kanayama, A. Nilpour, and C. A. Lelm, "A locomotion control method for autonomous vehicles," *IEEE Internat. Conf. Robotics and Automation,* Apr. 1988, pp. 1315–1317.

[4] T. Tsumura, N. Fujiwara, T. Shirakawa, and M. Hashimoto, "An experimental system for automatic guidance of robot vehicle following the route stored in memory," in *Proc. 11th Internat. Symp. Industrial Robotics,* 1981, pp. 187–193.

[5] M. R. Kabuka and A. Arenas, "Position verification for mobile robot using standard patterns," *IEEE Trans. Robot. Automation,* vol. 3, pp. 505–516, Dec. 1987.

[6] J. H. Kim and H. S. Cho, "Real-time determination of a mobile robot's position by linear scanning of a landmark," *Robotica,* vol. 10, no. 4, pp. 309–319, Jul./Aug. 1992.

[7] T. Hongo, H. Arakaw, G. Sugimoto, K. Tange, and T. Yamamoto, "An automatic guidance system of a self-controlled vehicle," *IEEE Trans. Industrial Elec.,* pp. 5–10, vol. 34, 1987.

[8] J. L. Crowley, "Dynamic world modeling for an intelligent mobile robot using a rotating ultrasonic ranging device," in *Proc. IEEE Internat. Conf. Robot. Automation,* pp. 128–135, Mar. 1985.

[9] D. Miller, "A spatial representation system for mobile robots," in *Proc. IEEE Intern. Conf. Robot. Automation.,* Mar. 1985, pp. 122–127.

[10] M. Drumheller, "Robot localization using sonar," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 9, no. 2, pp. 325–332, Mar. 1987.

[11] I. J. Cox, "Blanche—An experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. Robot. Automat.,* vol. 7, pp. 193–204, Apr. 1991.

[12] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Automation,* vol. 7, pp. 376–382, Jun. 1991.

[13] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox, "Dynamic map building for an autonomous mobile robot," *Inter. J. Robot. Research,* vol. 11, no. 4, pp. 286–298, Aug. 1992.

[14] A. Gelb (Ed.,), Applied Optimal Estimation. Cambridge: MIT Press, 1974.

[15] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* New York: Academic, 1970.

[16] R. K. Miller, S. B. Skaar, and W. H. Brockman, "Robust camera-space control of an automated forklift," in *Proc. 1991 AUVS Conf.,* Aug. 1991, pp. 295–307.

[17] S. B. Skaar, I. Yalda-Mooshabad, and W. H. Brockman, "Nonholonomic camera-space manipulation," *IEEE Trans. Robot. Automation,* vol. 8, no. 4, pp. 464–479, Aug. 1992.

[18] S. B. Skaar, W. Z. Chen, and R. K. Miller, "High-resolution camera-space manipulation," in *Proc. 1991 ASME Design Automat. Conf.,* Miami, Florida, 1991.

[19] P. Liang, Y. L. Chang, and S. Hackwood, "Adaptive self-calibration of vision-based robot system," *IEEE Trans. Syst. Man, and Cybernetics,* vol. 19, pp. 811–824, Jul./Aug. 1989.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes.* New York: Cambridge Univ. Press, 1986.

[21] E. T. Baumgartner, "An autonomous vision-based mobile robot," Ph.D. dissertation, Dep. Aerospace and Mech. Eng., Univ. Notre Dame, Notre Dame, IN, November 1992.

Eric T. Baumgartner received the B.S. degree in aerospace engineering from the University of Notre Dame, Notre Dame, IN in 1988 and the Ph.D. degree in mechanical engineering from the University of Notre Dame, in 1992.

He is currently an Assistant Professor in the Department of Mechanical Engineering and Engineering Mechanics at Michigan Technological University, Houghton, MI. He is also a Visiting Faculty Research Associate in the School of Health and Rehabilitation Sciences at the University of Pittsburgh, Pittsburgh, PA. His research interests are in the areas of control systems technology, rehabilitation engineering, and design.

Steven B. Skaar received the A.B. degree from Cornell University, Ithaca, NY, in 1975 and the Ph.D. degree in engineering mechanics from Virginia Polytechnic Institute and State University, Blacksburg in 1982.

He is currently an Associate Professor in the Department of Aerospace and Mechanical Engineering at the University of Notre Dame, Notre Dame, IN. His current research interests include the development and testing of reliable strategies to control both holonomic and nonholonomic devices using vision. Additionally, he does research related to the control of distributed elastic systems.